COMP3900 - Computer Science Project Report

Project Title: Level Up
Group Name: FAADS
Submission Date: 18 November 2021

z5207213@ad.unsw.edu.au Daniel Nguyen, z5207213, Developer
z5122503@ad.unsw.edu.au Fiona O'Chee, z5122503, Developer
z5349312@ad.unsw.edu.au Ali Khosravi, z5349312, Developer
z5186264@ad.unsw.edu.au Yiyun Mao, z5186264, Developer
z5258021@ad.unsw.edu.au Arthur Fung, z5258021, Scrum Master

Table of Contents

Background

Trying to find games that suit your tastes have always been difficult. Countless hours spent viewing online reviews and word of mouth has set plenty of expectations, only for the games itself to disappoint greatly in the end.

Level Up aims to solve this problem with our online e-commerce marketplace, with an integrated recommender system. Users are able to keep track of what games they want to play, they have played and those that they have enjoyed. The recommender system will take into account their preferences and personalise a list of games for them to enjoy. The preferences of other users with similar tastes in games are also taken into account by the recommender system and a separate list is made for the user.

The marketplace also has a social system, where users can add others as friends and buy them games from their wish list. Users are incentivised to actively participate in purchasing and reviewing games, with an in-built achievement system that will reward users for their interactions. These rewards can range from simple icons and banners on their profile, to special deals and discounts once they've hit certain milestones. Details such as rating and how many people have enjoyed the game will also be visible.

Speaking of profiles, users will have access to a list of games that they own, allowing them to download whatever game they want and for them to ultimately revisit their ratings and recommendations if they wish to do so.

Our Product

The emphasis on the visibility of the recommender system, along with the rating of the game, is to ensure the user wastes no time finding a game that they'll love. Other competing platforms, such as Steam (Valve Corporation) and Epic Games Store (Epic Games), organise their homepage in such a way that users are forced to look through as many games as possible before being able to narrow down their tags and preferences, in order to maximize profits. We aim to be as user-friendly as possible by only showing our users games that they would like immediately.

Our social friends feature also plays a big part within the community. We are able to easily see our friend's wish lists and their preferences are taken into account by our recommender system. In contrast to Epic Games Store (Epic Games) and Ubisoft Connect (Ubisoft), where their social systems are limited to adding other users to a friends list, we take our platform one step further and enable users to buy games for others from their wish lists.

Continuing on the focus on being a user-friendly platform, our rewards system incentivises users to purchase and review games for special deals in the form of cashbacks and coupons. Compared to the aforementioned gaming platforms, where deals happen monthly every few months, continuously using our platform will enable users to access deals every few days to weeks.

Novel Functionality

Our novel features come in the form of our:

1. Social friends system
   - This system lets users add others as friends
   - Let's users browse their friends wishlists
   - Integrates with the recommender system in order to provide game recommendations
   - Let's users buy their friends games from their wishlists

2. Rewards system
   - This system rewards experience points (XP) for using the platform
   - Actions such as purchasing and reviewing games give users XP in the form of achievements
   - XP is then accumulated into a user's level/badge
   - The higher a user's level/badge, the better their rewards
   - Rewards come in the form of a cashback

Overview - Architecture/design of the overall system

- **Architecture diagram**
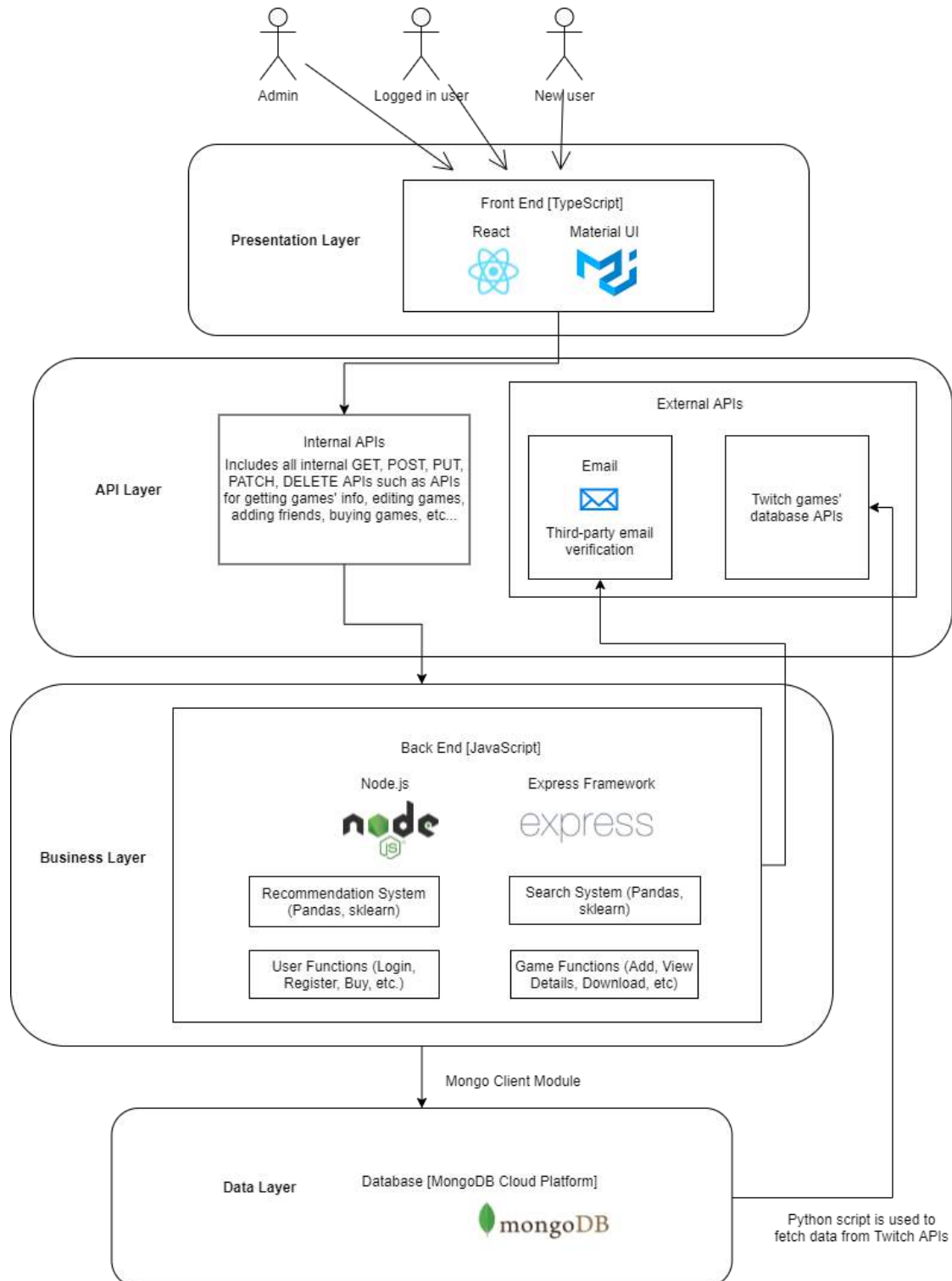
*Notes/References:
- The IGDB API (Twitch 2021) is used to populate games data.
- Nodemailer (Nodemailer 2021) is used to send verification emails to users.

Architecture Description:

1. Presentation Layer:
   The presentation layer is developed with the React.js framework alongside the additional ui library named Material-UI. The programming language used here is TypeScript.
   This layer provides admin, logged-in users and new users with user-friendly interfaces to interact with our platform, and it takes those users' inputs to our business layer using our internal API endpoints (in API layer) for further manipulation.

2. API Layer:
   This layer consists of internal and external API endpoints. Internal API endpoints are used for the connection between our presentation layer and business layer. On the other hand, external API endpoints are used for third party email verification (in registration and forget password), as well as feeding our game database using data available for games on the IGDB database.

3. Business Layer:
   The business layer is developed using Node.js and Express Framework, plus the modules pandas and sklearn in Python for the recommendation algorithms. The programming languages used are Javascript and Python.

   It takes the input from our presentation layer (using our API layer) and manipulates data for the data layer. Functions in this layer contain the recommendation and the search system (based on pandas and sklearn), functions for users including login, registration, buying games, and functions for e-commerce games such as adding/viewing game details and downloading new games.

   The third-party email is also used here for account verification.

4. Data Layer:
   The data layer is generated with the help of MongoDB, which is a cloud platform for databases. All data related to the platform is collected here and it provides data for the business layer so that we can manipulate that data and present it to the users. Our game database is filled using mongoose via the data provided by IGDB API endpoints.

- Use case diagram

* The database participates in every case, so it has been left out for simplicity

* Admin inherits user functions
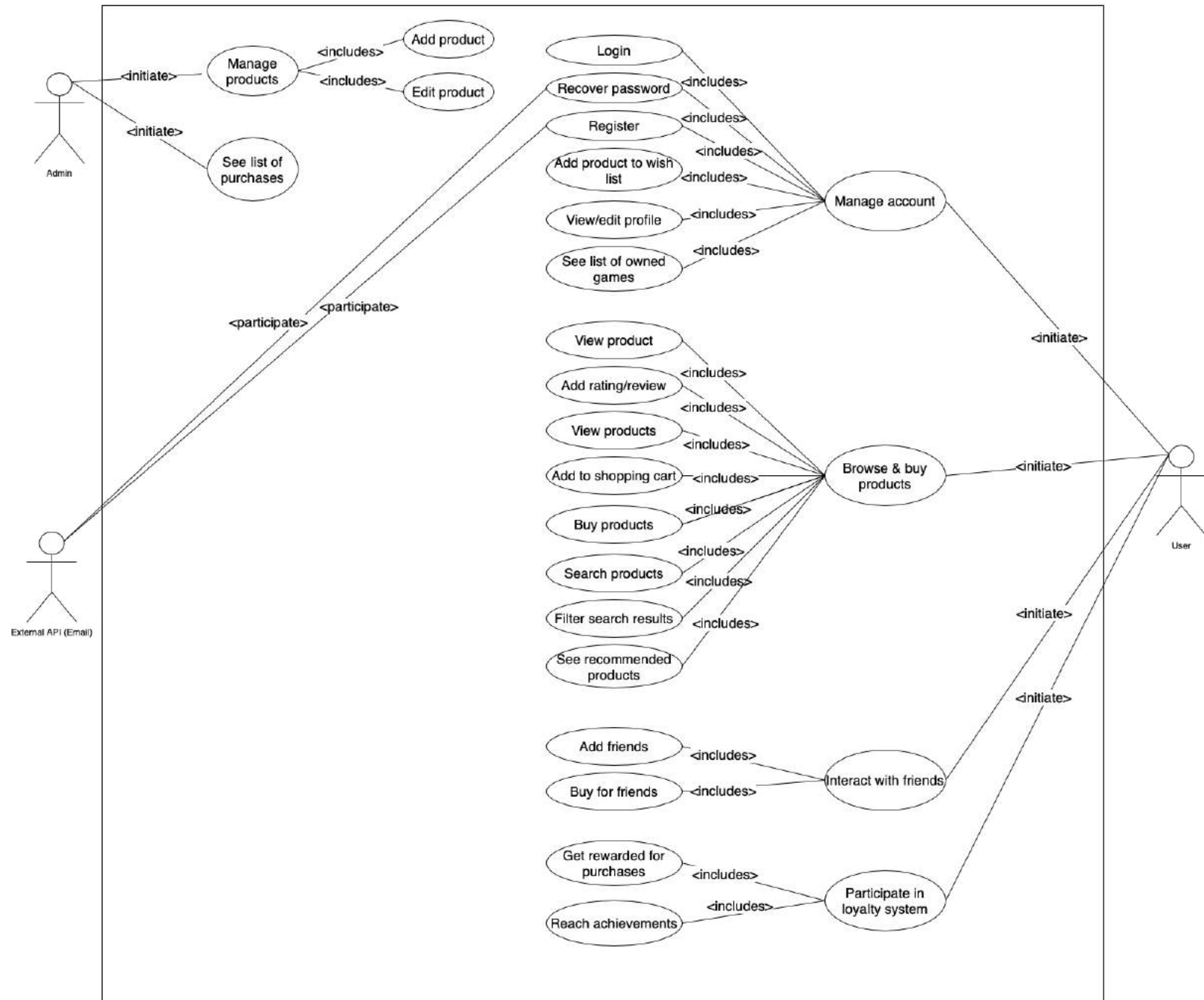
Actors

1. Admin:
   The administrator of the platform who owns all the games and earns money by selling them. The admin is also in charge of adding/editing game details and keeps track of all history in the platform.

2. Users:
   The users of the platform are those who visit our website, they should have a user account to log in, so that they can access all features that we provide for them to search/view/purchase/comment/download the games in our ecommerce platform.

3. External API:
   This is a third party API which provides us with some initial data on game details that we are trying to sell on the ecommerce platform. In addition, we used external APIs for sending emails in verification parts such as registration and forgetting password.

Functionalities

Epic story 1: Basic account setting - Login / Register Function

1.1 As a user/admin, I want to be able to register an account so that I can login to the platform

Acceptance Criteria:
- 'Sign Up/Register' button on the login page that redirects users to the registration page.
- On registration page:
    - Sign up by entering email address, username, password then submitting.
    - Can click button to redirect to login page.
- Email and username must be unique and email should be valid format.
- Password must be least 8 characters and include at least 1 uppercase, 1 lowercase, 1 number
- Email confirmation after sign up.
- Clicking email confirmation confirms account, then can redirect to login page.
- Password must be encrypted in the database.

Priority: 5/5
Difficulty: 3/5

1.2 As a user/admin, I want to be able to log in to an existing account so that I can access the platform as the associated user

Acceptance Criteria:
- There should be a 'Login' button on the top of each page, which redirects users to the login page.
- On login page:
    - Users can login with their email and password.
    - Can click 'Sign Up/ Register' Button to redirect to registration page.
- If the username has not been registered yet, show a message with "Account does not exist, please register to log in."

- Once the user successfully logs in, the user should be redirected to the home page.

Priority: 5/5
Difficulty: 2/5

1.3 As a user/admin, I want to recover my account so that I will be able to change my password and access my account even if I forget my password.

Acceptance Criteria:
- On the login page, there should be a 'forgot my password' button that redirects users to find their password.

- On the 'find my password' page, a form that asks users for their email address and the 'Send Me Code' button.
- If the email is invalid / not registered with system, show a message with "Account with this email address does not exist, please check details."
- If the email is registered, then an email will be sent to the given email address, containing the reset link.
- Users will need to input a new password:
    - Password must be least 8 characters and include at least 1 uppercase, 1 lowercase, 1 number

Priority: 1/5
Difficulty: 3/5

Epic story 2: Add / Edit / View Product Details
2.1 As an admin, I want to be able to add products to the store so that I can sell them.

Acceptance Criteria:
- Admins can add games with name, image, creator of game, description and price to the store.
- Price must be a valid number (max 2 decimal points)
- Admins can attach tags (i.e. genre of the game, themes in the game) to the store.
- Added products will be shown in the store.

Priority: 5/5
Difficulty: 3/5

2.2 As an admin, I want to be able to edit products in the store, so that I can reflect any product or stock changes in the store.

Acceptance Criteria:
- Admins can edit game name, image, creator of game, description and price.
- Price must be a valid number (max 2 decimal points).
- Edit history will be retained (both admins and customers can view).
- Admins can remove items from the store.

Priority: 3/5
Difficulty: 2/5

2.3 As a user, I would like to view product details, so that I can make a decision on whether the product interests me.

Acceptance Criteria:
- Users and admins can view product details, including title, description, game creator, image and price.

- If a game is already owned, it will say so on the page, and there will also be a download button on the product page.

Priority: 3/5
Difficulty: 2/5

2.4 As a user, I would like to be able to add ratings and leave reviews, so that I can share my thoughts on whether I liked the game or not.

Acceptance Criteria:
- Users will be able to leave reviews on games that that they own.
- Users and admins can view average ratings.
- Users and admins can view item ratings and reviews.

Priority: 1/5
Difficulty: 2/5

2.5 As a user, I would like to see the amount of people who have purchased an item, so that I can see how popular the game is.

Acceptance Criteria:
- Users can view people and amount of people who have purchased items.
    - Admin can see all usernames.
    - Username will be partially censored if the user is not a friend.

Priority: 2/5
Difficulty:1/5

2.6 As a user, I want a list of products to be shown on my homepage, so that I can browse items and click into products to see more details of each them.

Acceptance Criteria:
- Dashboard / home page will show products in the store.
- User will be able to click into each product to be directed to the individual product page, which will have more information on the individual game.

Priority: 5/5
Difficulty: 3/5

Epic story 3: Shopping Cart / Purchase System

3.1 As a user, I would like to be able to add products to a shopping cart, so that I can browse and add multiple items that I would like to buy together easily.

Acceptance Criteria:
- Users can add items to a shopping cart.
- Users can view their shopping cart, and edit/remove items from it.
- Users can view the total price of items in the shopping cart.

Priority: 5/5
Difficulty: 2/5

3.2 As a user, I would like to checkout items in my shopping, so that I can purchase items that I want.

Acceptance Criteria:
- Users can checkout items in their shopping cart.
- Checks if there are enough store credits in their account, otherwise errors.
- Users can pay using credit card and/or coupon if insufficient credits
- Credit card will check for 16 digits, expiration date and 3 digit CVV
- Sends email notification on purchase.
- Users can download games after purchase.
- Download page (after purchasing) has a download button next to each game that has just been purchased.

Priority: 5/5
Difficulty: 3/5

3.3 As a user, I want to be able to add items I'm interested in buying to a wish list, so that I have a plan of what I would like to buy in the future.

Acceptance Criteria:
- Users can add products to their wish list.
- Users can view products in their wish list and edit their wish list (remove items from it) on their profile.
- Once a user has obtained (by buying or being gifted) an item on their wishlist, it will be automatically removed from their wishlist.
- Users can move the product from their wishlist to shopping cart.

Priority: 3/5
Difficulty: 3/5

Epic story 4: Profiles

4.1 As a user, I want to have a centralised location where I can view and edit my account details, so that I can check them and keep them up to date.

Acceptance Criteria:
- Users can edit their linked email, username, password and display pictures on their profile.
- Users can view their username, email (censored after the first letter and before the @) and display pictures on their profile.

Priority: 3/5
Difficulty: 2/5

4.2 As an admin, I want to be able to see a list of purchases so that I can keep track of what products are sold.

Acceptance Criteria:
- Admins can see the history list of purchases in the store.
- Admins can use a search bar to filter through the history.
- Admins can see the reviews and ratings associated with each purchase.
- Admins can select to filter which fields to display (i.e. if they only want to see ratings)

Priority: 2/5
Difficulty: 3/5

4.3 As a user, I want to have a clear monetary system, so that I can purchase items on the site.

Acceptance Criteria:
- Users can redeem store credits in their profile
    - Need to pass captcha to do so
- Users also receive email confirmation (don't need to click any verification, just to notify that they have purchased from the store).
- Users can view their store credit balance in their profile.
- Admins can withdraw store credits
    - Need to pass captcha
    - Will receive email confirmation afterwards.

Priority: 5/5
Difficulty: 3/5

4.4 As a user, I want to be able to see a list of the games I own, so that I can redownload them if I need to.

Acceptance Criteria:
- Show list of owned games in the user's profile.
- There will be a download button next to each game.

Priority: 2/5
Difficulty: 1/5

Epic story 5: Search System

5.1 As a user I want to search products based on keywords to locate specific games I'm interested in.

Acceptance Criteria:
- Autocomplete results should appear as the user types
- Hitting enter or pressing the search icon should bring up search results.

Priority: 5/5
Difficulty: 4/5

5.2 As a user I want to be able to filter search results so that I can categorize search results.

Acceptance Criteria:
- Range of filters on the left side of the screen.
- Selecting a filter should automatically filter the results.
- Filters to include are by rating, genre and cost.

Priority: 3/5
Difficulty: 2/5

Epic story 6: Recommender System

6.1 As a user, I want the system to recommend products similar to my search history to me so that I can find my favorite products faster.

Acceptance Criteria:
- Upon opening the dashboard / home page by users, a list of recommended products are shown.
- Recommended items have similar tags, based on genres and themes, to existing items in the user's search history.
- New searched items are added to the user's search history and will be used in the future.

Priority: 5/5

Difficulty: 4/5

6.2 As a user, I want the system to recommend products bought by people with similar opinions about products to me so that I can find my favorite products faster.

Acceptance Criteria:
- Upon opening the dashboard / home page by users, a list of recommended products are shown.
- Recommended items are based on activity of other users who have similar likes and dislikes towards certain games as me.
- New ratings of a user for a product will be used to recommend products to other users.

Priority: 4/5
Difficulty: 5/5

Epic story 7: Social 'Friends' Feature
7.1 As a user, I want to be able to search other users by username so that I can add them as my friends.

Acceptance Criteria:
- A search bar should be provided to search for other users via username.
- By default, users can only view the username and profile image of others.

Priority: 3/5
Difficulty: 2/5

7.2 As a user, I want to be able to add other users as friends so that I can view their activities or wishlists.

Acceptance Criteria:
- Users can only use this function when successfully logged in.
- Users can request to add another user as a friend which will automatically send a notification to the user.
- The user who has been added can then accept or decline the friend request.
- Once the friend request is accepted, users will be able to view their owned games or wishlists.

Priority: 3/5
Difficulty: 4/5

7.3 As a user, I want to be able to purchase games for others so that I can buy gifts for my friends based on their wishlist.

Acceptance Criteria:

- If users are purchasing games for their friends, a notification containing a link to the game listing will be sent to their friends.
- Their friends can download games via the link (as it is now an owned game).

Priority: 1/5
Difficulty: 4/5

Epic story 8: Rewards System
8.1 As a user I want to be rewarded for spending money so that I am incentivized to purchase products

Acceptance Criteria:
- Users get store credit for every $100 they spend. The amount received depends on current badge (when level benchmark achieved) from loyalty system.
    - $5 every $100 for bronze (lvl 1)
    - $10 every $100 for silver (lvl 5)
    - $15 every $100 for gold (lvl 10)
    - $20 for every $100 for platinum (lvl 20)
    - $25 for every $100 for diamond (lvl 40)
- Users will receive a snackbar notification signifying they have received the reward.

Priority: 3/5
Difficulty: 2/5

8.2 As a user I want to be able to reach achievements so that I can enjoy purchasing products more.

Acceptance Criteria:
- Users can view all achievements on their profile page.
- When an achievement is reached, a snackbar notification will show they have done so.
- Achievements grant exp.
- Achievements:
    - Buy x amount of products
    - Spend x amount of money
    - Gift x amount of products

Priority: 3/5
Difficulty: 4/5

8.3 As a user I want a loyalty system so that I am incentivized to purchase on this site.

Acceptance Criteria:
- The loyalty system will consist of a profile's level.

- User's earn experience for purchasing products (money spent) and reaching achievements.

Priority: 3/5
Difficulty: 3/5

How our functionalities have addressed all project objectives

Objective 1: Users must be able to create an account which they can use to login and edit account details when they want to.

- **Referring to user story 1.1**, users will be able to register an account securely, with password complexity requirements, email confirmation and encryption of user's stored passwords.

Users are able to create an account, with password complexity requirements of at least 8 characters, 1 uppercase letter, 1 lowercase letter and 1 number. The email is required to be of valid format, the username must be 3-18 characters in length, without spaces, and the email and username must not be the same as that of an existing account. Users must also verify their account, using their email, before they can log in. The verification link expires after 10 minutes. The password is encrypted using 256-bit AES encryption on the server side.

```
password: CryptoJS.AES.encrypt(
  password,
  process.env.AES_KEY
).toString(),
```

This satisfies the part of the objective where users must be able to create an account.

- **Referring to user story 1.2**, users will be able to login using the email and password of a registered account.

Users can login, using the email and password of a registered and verified account. If the account has not been verified, they will be prompted to verify their account. This satisfies the part of the objective of users being able to log in.

- **Referring to user story 1.3**, there will also be a password recovery option, which can be used to recover the password of an account if the user has access to the associated email address.

Users are able to recover their account if they forget their password, by receiving a link in their email to change their password.

- **Referring to user story 4.1**, users will be able to view their account details (username, censored email and password) on their profile, and edit these when they want to.

Profile

Username: fifi   ✎

Email: f***********@gmail.com   ✎

Password: ************   ✎

Users are able to view their account details on their profile. The email and password are censored for security purposes. Users can edit their username, email, password and profile image, using the edit buttons., thus fulfilling the part of the objective for users to be able to edit account details.

Objective 2: Admins must be able to add games to the store, with game details, edit them to update product details, and view their purchases and profits.

- **Referring to user story 2.1**, admins will be able to add games to the store, with the game details including name, image, creator, description, price and genre tags.

Add Game Page:
1. Games can be added using this interface by the admin, which is directed by the "ADD GAME" button on the top bar.
2. There are two sections in this interface, the preview and the edit game section.

3. Preview:
   This is a preview of how your game information section would be shown if it is uploaded.
4. Edit game section:
   a. Inputs are present for setting game name, game summary, price, systems, companies, genres and the game image. Input details:
      i. The game name and game summary are input as text.
      ii. The price is input as a number with two decimal digits allowed.
      iii. Systems, companies and genres are to be selected, and users can also input new companies and new genres that are not on the select options.
      iv. The image can be uploaded as a file.
      v. Once the "ADD GAME" button is clicked, the new game would be generated. And a success message would be shown once succeeded.

   This is how we satisfy the "add games to the store" part of our objective.

- **Referring to user story 2.2**, admins will be able to edit details of games in the store. Edit history will be retained for store transparency.

As we can see here, admins are able to edit a game's details, with the current details being autofilled so that no time is wasted copy-pasting any sections that do not need to be edited. The edited sections are also previewed at the top in real time so that admins will know what it'll look like to users to prevent any formatting mistakes. Clicking on the "show edit history" button will lead us on to the next part of this user story.

Edit History

SHOW EDIT GAME

Date	Field	Old Value	New Value
12/11/2021, 3:04:00 pm	Name	Overlord: Dark Legend Version 10	Overlord: Dark Legend Version 11
	Price	210	100
	Summary	Dark Legend is a spinoff of the 2007 video game Overlord, and is set as a prequel. The gameplay is similar to Overlord, but presents some differences. While the game does not have several features of the original game or its sequel, Overlord II, it offers some new content and a new gameplay style taking advantage of the Wii controller. Reviews of the game are mixed: praising the humour, story, and controls of the game, but finding problems with the animation, short game length, and lack of complexity or difficulty.	For testing purpose
12/11/2021, 2:46:34 pm	Name	Overlord: Dark Legend	Overlord: Dark Legend Version 10
	Price	146	210
	Summary	Dark Legend is a spinoff of the 2007 video game Overlord, and is set as a prequel. The gameplay is similar to Overlord, but presents some differences. While the game does not have several features of the original game or its sequel, Overlord II, it offers some new content and a new gameplay style taking advantage of the Wii controller. Reviews of the game are mixed: praising the humour, story, and controls of the game, but finding problems with the animation, short game length, and lack of complexity or difficulty.	Dark Legend is a spinoff of the 2007 video game Overlord, and is set as a prequel. The gameplay is similar to Overlord, but presents some differences. While the game does not have several features of the original game or its sequel, Overlord II, it offers some new content and a new gameplay style taking advantage of the Wii controller. Reviews of the game are mixed: praising the humour, story, and controls of the game, but finding problems with the animation, short game length, and lack of complexity or difficulty.

Admins can view the edit history of a game, being shown the date of edit along with the pre and post edit values. Admins can then click the "show edit game" button to be shown the original edit page, thereby satisfying the objective of "edit to update product details".

- **Referring to user story 2.5**, admins will be able to view the the complete usernames of the people who have purchased a game

Game Information

STAR WARS EMPIRE AT WAR

Star Wars: Empire At War

From the lives of soldiers to the deaths of planets, you are the supreme galactic commander. It is a time of galactic civil war. Take up the reins of the Rebellion or assume control for the Empire. Whichever you choose, it will be up to YOU to steer your side to ultimate victory. Command everything from individual troops to starships and even the mighty Death Star as you execute campaigns on the ground, in space and across the galaxy. Forget tedious resource gathering – just jump straight into the heart of the action. You can even change Star Wars history! Every decision affects your next battle and every battle helps shape the fate of the galaxy.

RELEASE DATE: 2006-02-16

DEVELOPER: HiRabbit
James Zaiewski

GENRE: Real Time Strategy (RTS)
Strategy

★★★

$77   EDIT   ADD TO CART

Reviews

No Comments Yet!

Friends Who Play This Game (0)

Others Who Have Bought This Game (4)

Test1@gmail.com
Test2@gmail.com
Testing1234
my1234

Compared to users, who will only be shown the full names of users that are their friends, or have the names censored otherwise, admins are able to view the complete, uncensored list of users who have bought a game.

- **Referring to user story 4.2**, admins will be able to view a history list of purchases from their store on their profile, including profits, ratings and reviews.

Purchase History

Client	Game Purchased	Price	Date Purchased
Rhea	Squid Game - dalgona	$120.00	18/11/2021, 10:35:39 am
Judyyy	Silent Hunter III	$120.00	18/11/2021, 10:35:39 am
DorseyGodines	Overlord: Dark Legend Version 11	$100.00	15/11/2021, 10:57:24 pm
DorseyGodines	Aion	$113.00	15/11/2021, 10:01:10 pm
my1234	Aion	$113.00	13/11/2021, 4:53:25 pm
Testing1234	Aion	$113.00	13/11/2021, 4:32:57 pm
AdminOne	Silent Hunter III	$120.00	12/11/2021, 9:29:53 am
AdminOne	Overlord: Dark Legend Version 11	$100.00	12/11/2021, 9:29:53 am
aaaaa	Silent Hunter III	$120.00	12/11/2021, 9:29:53 am
aaaaa	Overlord: Dark Legend Version 11	$100.00	12/11/2021, 9:29:53 am
fhchx	Silent Hunter III	$120.00	12/11/2021, 9:29:53 am
fhchx	Overlord: Dark Legend Version 11	$100.00	12/11/2021, 9:29:53 am
asdsada	Silent Hunter III	$120.00	12/11/2021, 9:29:53 am
asdsada	Overlord: Dark Legend Version 11	$100.00	12/11/2021, 9:29:53 am
boriho2424	Silent Hunter III	$120.00	12/11/2021, 9:29:53 am
boriho2424	Overlord: Dark Legend Version 11	$100.00	12/11/2021, 9:29:53 am
boriho2424asda	Silent Hunter III	$120.00	12/11/2021, 9:29:53 am
boriho2424asda	Overlord: Dark Legend Version 11	$100.00	12/11/2021, 9:29:53 am
siegfried0	Silent Hunter III	$120.00	12/11/2021, 9:29:53 am
siegfried0	Overlord: Dark Legend Version 11	$100.00	12/11/2021, 9:29:53 am

Rows per page: 5 ▾   1-20 of 61 |<   <   >   >|

Admins are able to view the site's purchase history. The values shown are: The user who purchased the game, the game purchased, its price and the date purchased. The admin can then click on the game to be taken to the store page, where the ratings and reviews are displayed.

Objective 3: Users must be able to view games and game details in the store, with options to add the game to their cart or to their wishlist.

- **Referring to user story 2.3**, users will be able to view games and details of each game, including name, image, creator, description and and price.

Game Information Page:
All information about a certain game is presented here with its game name, image, creator companies, game description and the price.
This is how we satisfy the "game details" part of our objective.

- **Referring to user story 2.4**, users will be able to view ratings and reviews, to help with making a decision on whether or not to purchase the game.

Reviews

★★
OK

★★★★★
Love it!

Reviews

LEAVE A COMMENT

★★★★
Fun!

★★★★★
Love it!

★★
testing

★
Bad
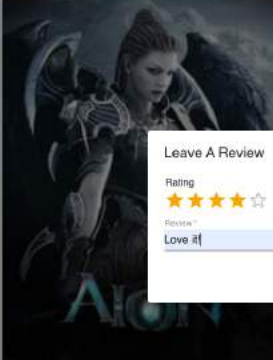
★★★★
Love it!

Game Information

Aion

A massively multiplayer online role-playing game released by NCSOFT. The game combines PvP and PvE in a fantasy game environment.

RELEASE DATE:

2009-07-17

Nintendo EAD Group No. 3

Serafian

Role-playing (RPG)

Turn-based strategy (TBS)

Leave A Review

Rating
★★★★☆

Review *
Love it!

CANCEL   SUBMIT

$113

DOWNLOAD

Reviews

LEAVE A COMMENT

★★★★
Fun!

Ali.Kh

Review Section in Game information Page:
All users can view the reviews and ratings for a certain game. In addition, if you have already owned the game, you can add a comment to the game.
This is how we satisfy the "rating and review" part of our objective.

- **Referring to user story 2.5**, users will also be able to see the amount of people who have purchased the game, to help with making a decision on whether or not to purchase the game.

Some of the game details objectives are achieved by viewing who has bought the game. As the user story describes, we can see who has purchased the game in addition to distinguishing if it is a friend who has purchased the game, or strangers via censored names.

- **Referring to user story 2.6**, users will be able to view a list of games on their homepage and click into one to see more detailed information about the individual game.
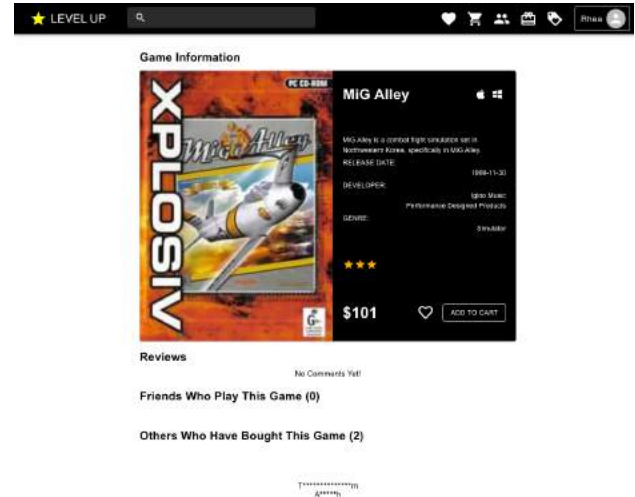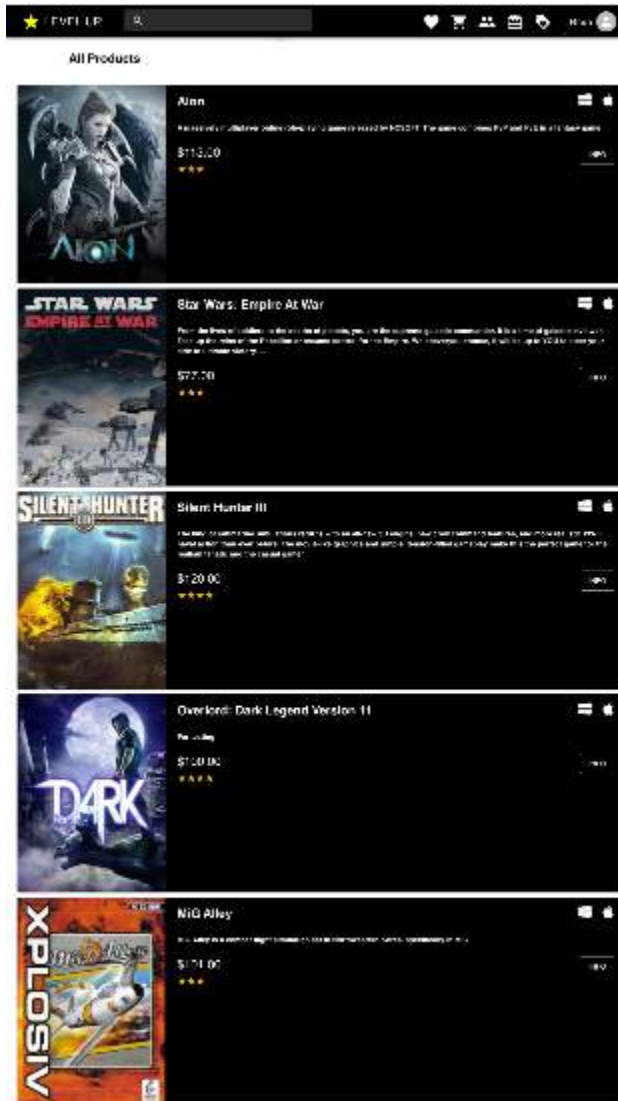
Homepage & link to Game Information Page:

As we can see on the left image, we can view all games in our homepage and redirect to the game information page, which is shown on the right hand side, by clicking the button "Info" on each of the game cards.

This is how we fulfill the feature that allows users to navigate to the game detail page of the game that they are interested in.

- **Referring to user story 3.1**, users will be able to add games to their shopping cart, view and edit or remove items in their shopping cart, and view the total price of all items in their shopping cart.

My Shopping Cart

Aion

A massively multiplayer online role-playing game released by NCSOFT. The game combines PvP and PvE in a fantasy game

$113.00
★★★

REMOVE FROM CART   INFO

Total Price: $113.00

REMOVE ALL   CHECKOUT

Users will be able to add games to their shopping cart. Here, the price of each game and the total price of the cart will be shown. They are able to remove games one by one, or remove the entire cart at once. It will also show the user if the game is part of their wishlist and give them the ability to add it to their wishlist if they wish to purchase it at a later time.

- **Referring to user story 3.3**, users will be able to add games to their wish list and view, edit or remove items in their wish list. They will also be able to directly add items from their wishlist to their shopping cart, so that they can buy the item.

Wishlist:

We can add a game into our wishlist by clicking the hollow heart icon on the game info card, and a success message will show up once succeeded, then we can see the game in our wishlist which is redirected by clicking the 'heart' icon on the top bar of our interface.

In the wishlist interface, we can remove the game from wishlist by clicking the filled 'heart' icon on each of the game cards, a success message will show up once succeeded. Also, we can add the game to the shopping cart by clicking the button 'ADD TO CART'.

This is how we satisfy the "wishlist" part of our objective.

Objective 4: Users must be able to purchase games and then download, as well as redownload games which they own.

- **Referring to user story 2.3**, there will be a download button on the game listings of the games which the user already owns.

Game Information

Reviews   LEAVE A COMMENT

No Comments Yet!

Friends Who Play This Game (0)

Others Who Have Bought This Game (0)

Once a user has purchased a game, the "add to cart" button will then be replaced with a "download" button so the user is not able to accidentally re-purchase any owned games.

- **Referring to user story 3.2**, users will be able to checkout items in their shopping cart using store credits, a credit card or a coupon. They will then be taken to a screen where they can download the games they have just purchased.

Users will be able to check out their shopping cart. They are given 3 options to choose from when it comes to payment, store credits, credit card and coupons. The store credits will check whether they have sufficient credits, the credit card will check for validity (16 digit card, expiry date and 3 digit CVV) and the coupon will deduct from the total purchasing price.

- **Referring to user story 4.3**, users will be able to redeem store credits by passing a captcha. They will receive an email notification of the purchase. Users will also be able to view their store credit balance in their profile.

Store Credits

$10.00

| ADD $10 CREDITS | ADD $20 CREDITS | ADD $50 CREDITS | ADD $100 CREDITS |

Confirm

Add 100 credits to account?

reCAPTCHA
☐ I'm not a robot
reCAPTCHA
Privacy - Terms

✕ CANCEL

Store Credits

$110.00

| ADD $10 CREDITS | ADD $20 CREDITS | ADD $50 CREDITS | ADD $100 CREDITS |

Confirm

Add 100 credits to account?

✓ I'm not a robot
reCAPTCHA
Privacy - Terms

✕ CANCEL   CONFIRM ✓

LA **LevelUp Admin**   12:44 pm
Add credits confirmation
To: fionaocheese@gmail.com

100 credits added to account.

As seen above, users can view store credit balance, and add store credits, by passing a captcha. They will also receive an email notification of their purchase. These credits can be used to purchase games, as per the project objective.

- **Referring to user story 4.4**, users will be able to see a list of games that they own, on their profile. There will be a download button next to each of the games they own so that they can redownload games which they own.

As seen above, users can see their owned games, and there is a filter bar which users can use to search through their owned games. They are able to redownload games that they own here, satisfying the redownloading part of the project objective.

Objective 5: Users must be able to find games which may interest them through store recommendations.

- **Referring to user story 6.1**, the store home page will show recommendations based on the user's search history and what they have viewed previously. It will match products based on items which have similar tags, such as genres and themes.
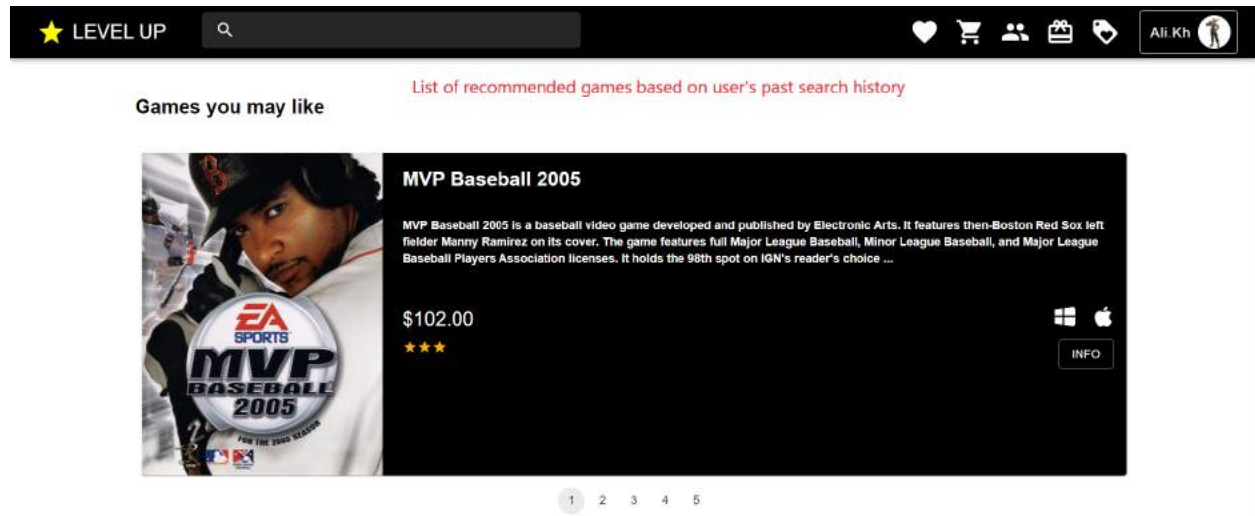
As it can be observed above, users can see a list of recommended games based on their past search history. Thus, the recommendation functionality is partially achieved.

- **Referring to user story 6.2**, users recommendations will also take into account the activity of other users who have similar likes, dislikes and ratings for games. New ratings of a user for products will be used to recommend products to other users as well.
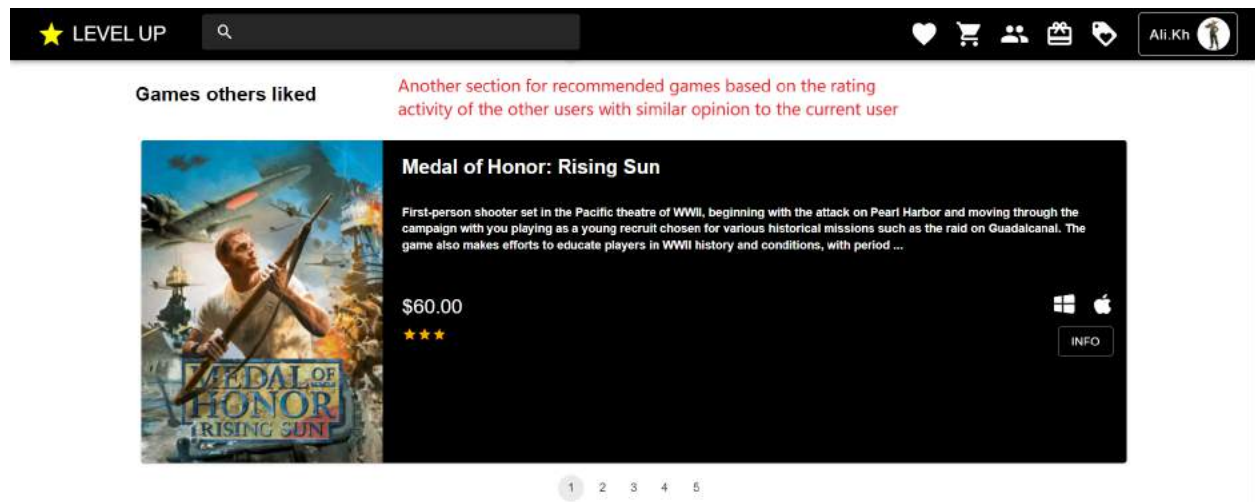
As it can be observed above, users (with at least one given rating) can see another list of recommended games based on the rating activity of the other users with similar rating activity. Thus, the recommendation objective is completed.

Objective 6: Users must be able search and filter through games in the store, to easily find games they are looking for, or would like to purchase.

- **Referring to user story 5.1**, users will be able to search products using a search bar to locate specific games that they are interested in. Autocomplete suggestions for possible search queries will be shown as the user types.

As it can be observed above, users can search games using the search bar and autocomplete suggestions are available below that. Thus, the search functionality objective is achieved.

- **Referring to user story 5.2**, users will be able to filter search results, using a range of filters, including average rating, and cost of the game.
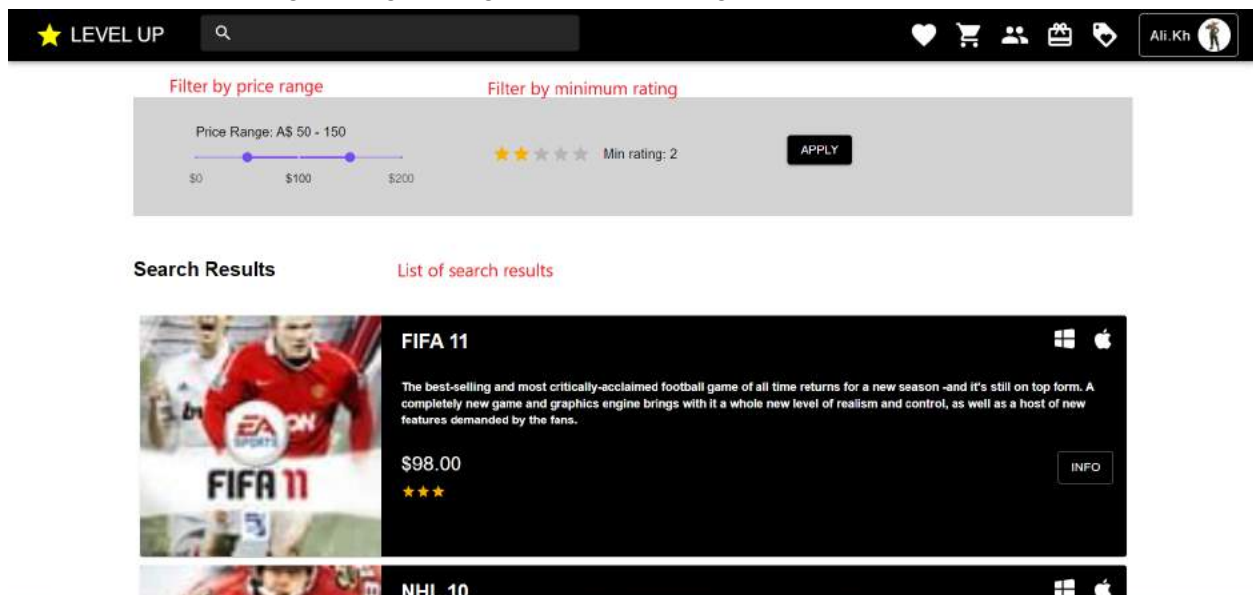
As it can be observed above, users can filter the search results by rating and cost. Thus, the search filter objective is achieved.

Objective 7: Users must be able to add friends and then purchase games that their friends want for them.

- **Referring to user story 7.1**, users will be able to search other users by username and add them as friends.
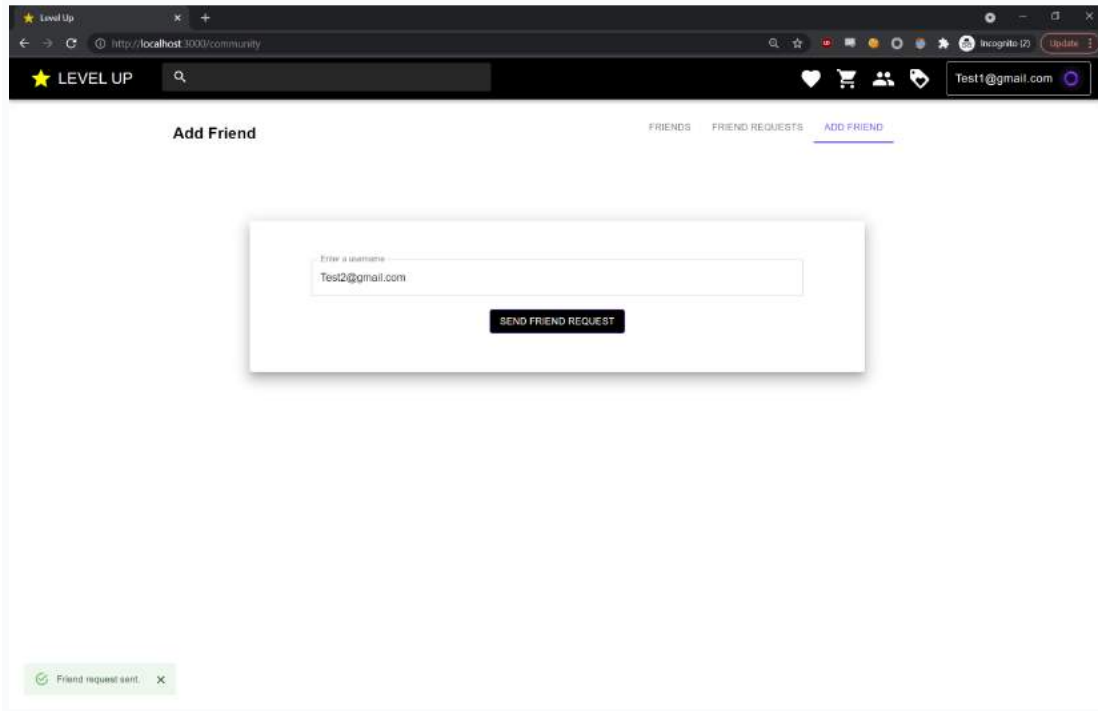
As you can see we can search a friend by username, and add them.

- **Referring to user story 7.2**, adding another user as a friend will automatically send another notification to the user. The user will then be able to accept or decline the friend request. Users will be able to view the owned games and wishlists of their friends.
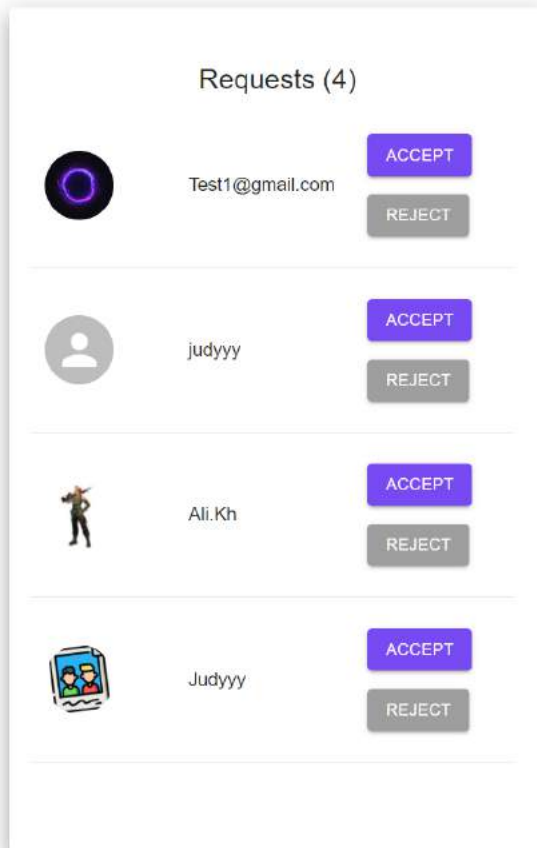
Above, we added Test2@gmail.com as a friend as Test1@gmail.com, so now that is visible as a friend request.
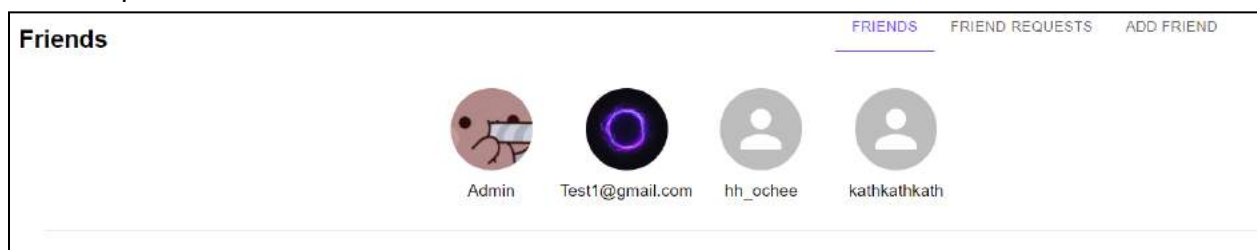
Pressing accept, we add them as a friend, thus adding them to our friends list and achieving that part of the objective.
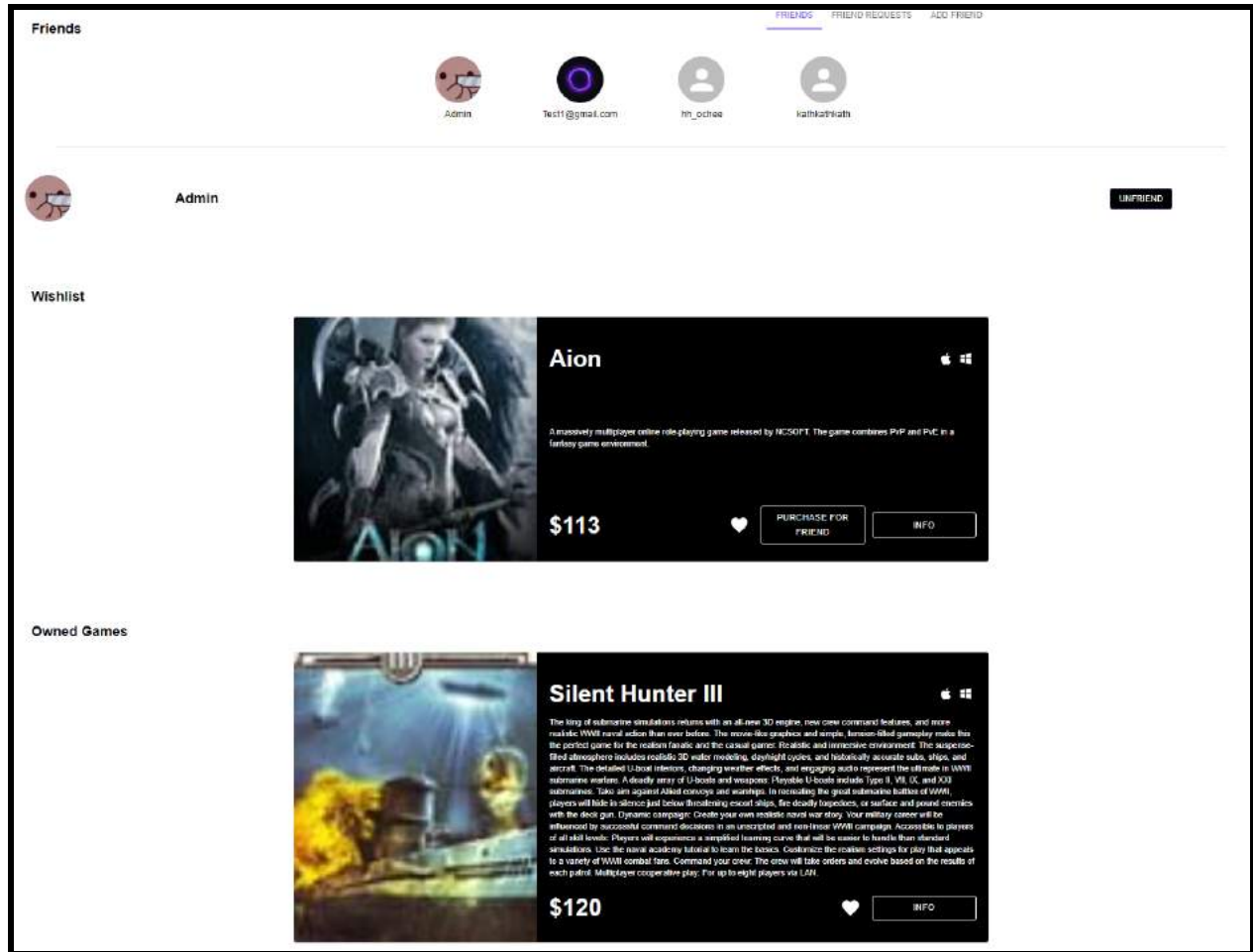
Here we can see our friend's wishlist and games, since they are our friends.

- **Referring to user story 7.3**, users will be able to purchase games for their friends. A notification containing the game listing, which will include the download button, will then be sent to the friend who is receiving the gift.
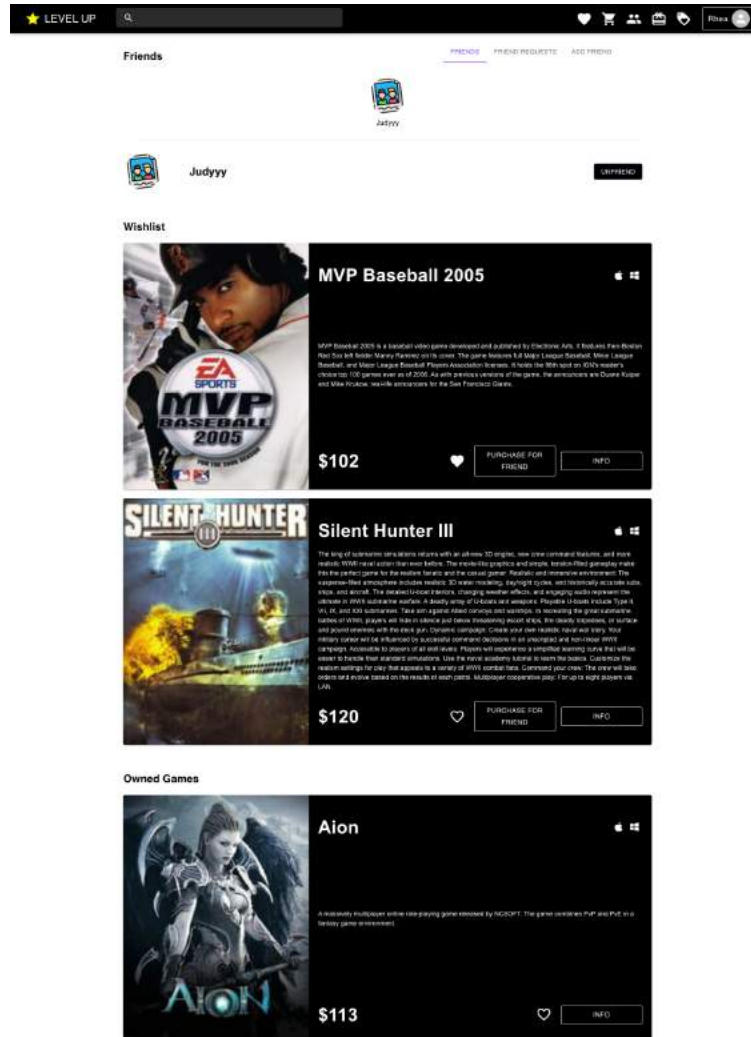
Gift for friends:

If a user wants to purchase a game for a friend, the user can view his/her wish list and list of owned games in the community by clicking the "community" icon on the top bar. Then the user can click the button "PURCHASE FOR FRIEND" on the game card which redirects him/her to the checkout page for bill-paying as in the following screenshot.

Once purchased, the user's friend can see the game in the Gift page by clicking the 'gift' icon on the top bar, it will show who the sender is and a button to "download".
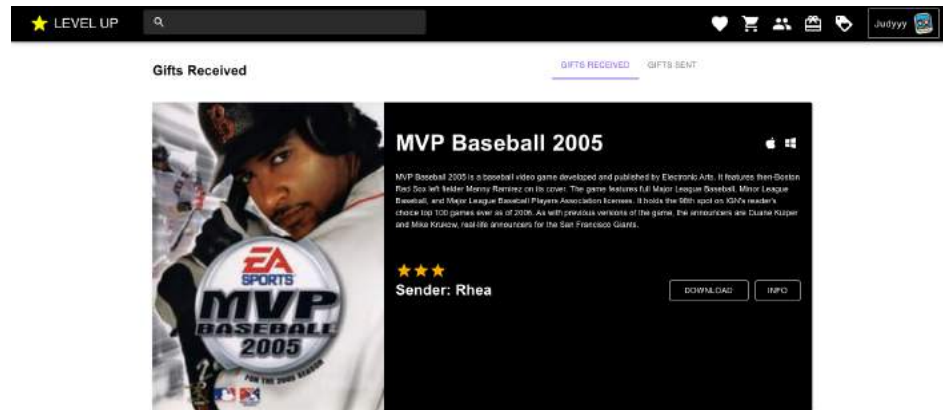
Also, the friend can view the game in the owned game which shows that he/she has already owned the game.
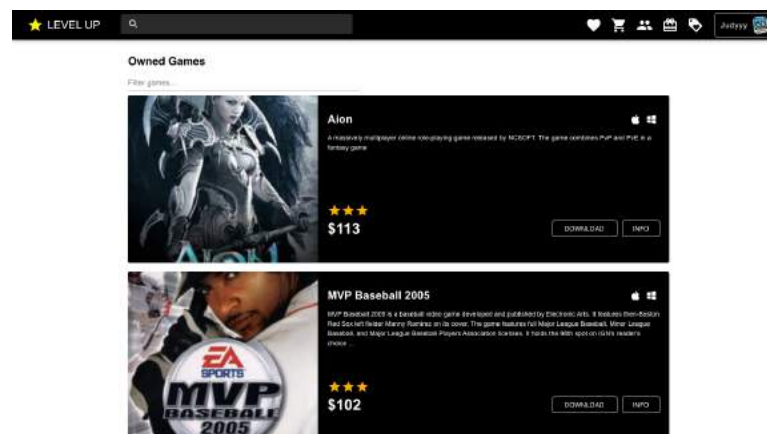
From the user's side, the user can see the game in the section 'gifts sent' in the gift page which would show the game receiver on the information card.
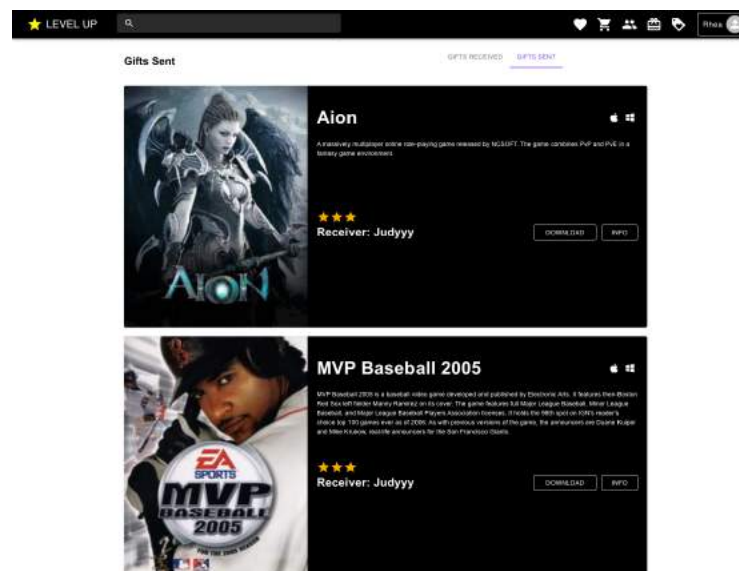
This is how we fulfil the "purchase for friends" part in our objective.

Objective 8: Users must be able to earn rewards, as well as level up with money spent, to incentivize purchases with the store.

- **Referring to user story 8.1**, users will get bonus store credits for every $100 they spend. The amount received depends on their current badge, and increases depending on the level benchmark achieved from the loyalty system. Users will receive a notification when they receive a reward.

Cashback Badge ⓘ

GOLD

User is currently a gold badge.

Reward Summary

$5 cashback every $100 spent for Bronze badge (lvl 5)

$10 cashback every $100 spent for Silver badge (lvl 10)

$15 cashback every $100 spent for Gold badge (lvl 20)

$20 cashback for every $100 spent for Diamond badge (lvl 40)

According to reward summary, he should earn $15 cashback for every $100 spent

Store Credits

$815.00

User currently has $815.

Your current store credit $815.00
Total: $140.00

USE STORE CREDIT   140|

User makes a 140 dollar purchase.

✅ 140 exp earnt! Levelled up to level 23! Cashback of $15 received!   ✕

Cashback notification is shown.

Store Credits

$690.00

User spent 140 dollars, so should have 675 left. Cashback of 15, so 690 should remain as seen.

Thus the rewards objective is completed.

- **Referring to user story 8.2**, users will be able to earn achievements from buying X amount of products, spending X amount of money, and adding X number of friends. Users will receive a notification and earn experience on reaching achievements. They will also be able to view all achievements on their profile page.

Achievements

50xp	Spending your first $100	100xp	Spending 500	200xp	Spending $1000
50xp	Buying your first game	100xp	Buying 5 games	200xp	Buying 10 games
50xp	Adding your first friend	200xp	Adding 5 friends	400xp	Adding 10 friends

This user currently has not achieved the achievement for buying their first game.

New achievement reached!

Dialog box after reaching a new achievement

New achievement reached for Buying your first game! 50 exp is granted as the reward!

OK
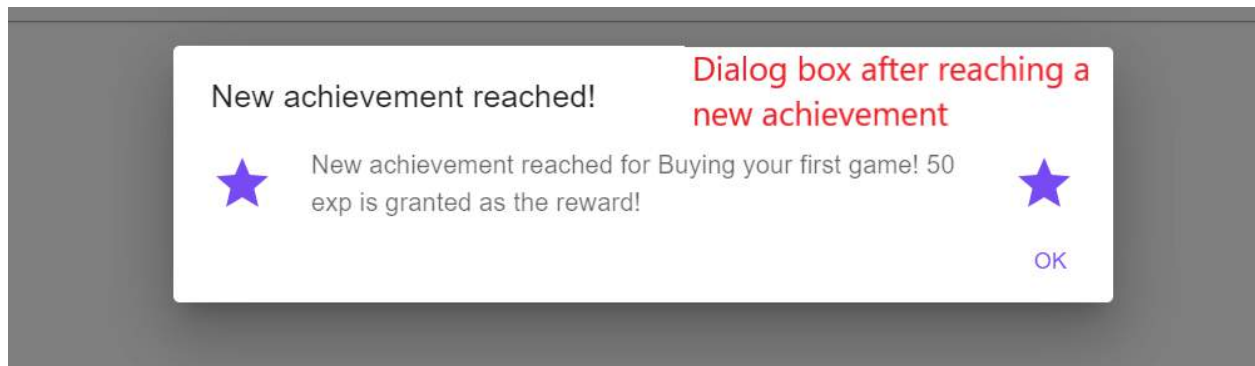
After buying their first game, a dialog box pops up showing that the user reached a new achievement.

Achievements

50xp	Spending your first $100	100xp	Spending 500	200xp	Spending $1000
50xp	Buying your first game	100xp	Buying 5 games	200xp	Buying 10 games
50xp	Adding your first friend	200xp	Adding 5 friends	400xp	Adding 10 friends

Now, the user can see the updated list of achievements. Thus, the achievement objective is completed.

- **Referring to user story 8.3**, the store loyalty system will consist of a user's level, which can be seen on their reward's page. User's will earn experience points for spending money and reaching achievements.

Rewards ⓘ

LEVEL 21

53/120

Cashback Badge ⓘ

GOLD

Here is our rewards display page, where we can track our levelling up.
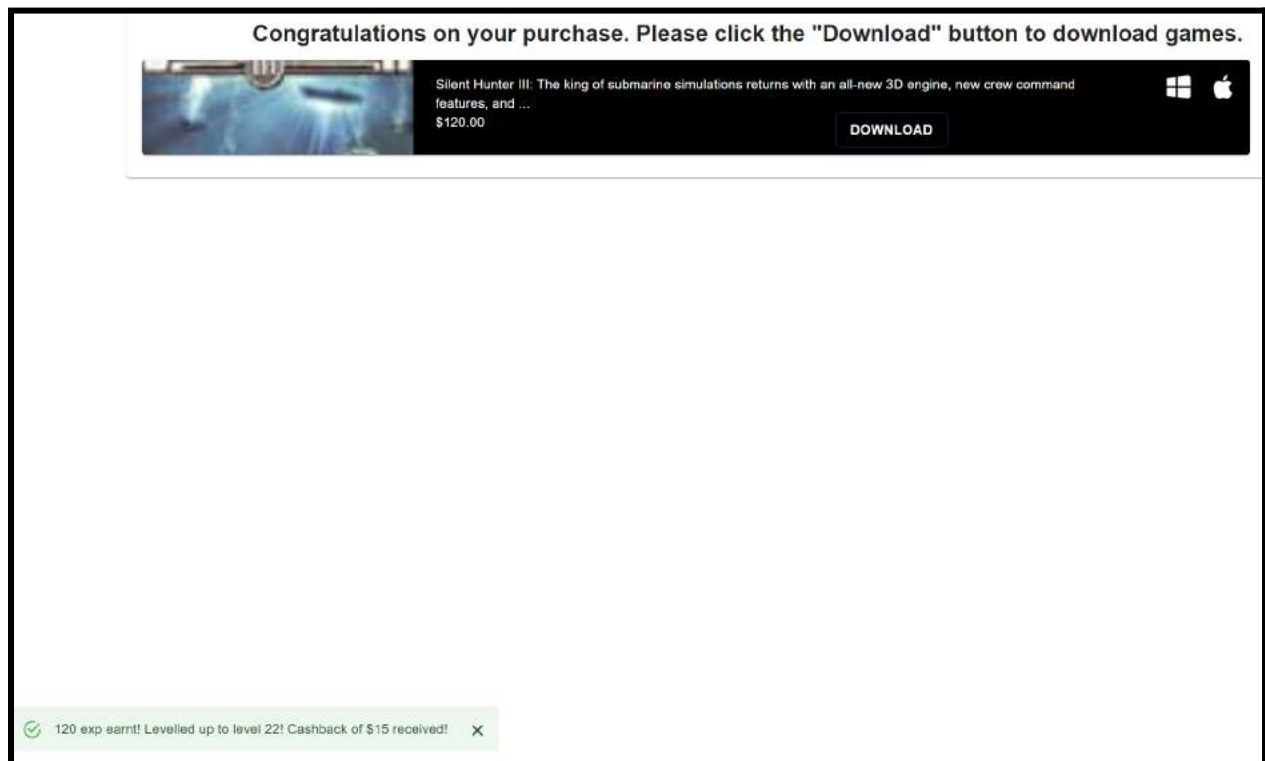
We have purchased a product thus earning 120 experience points as shown in the snackbar notification.
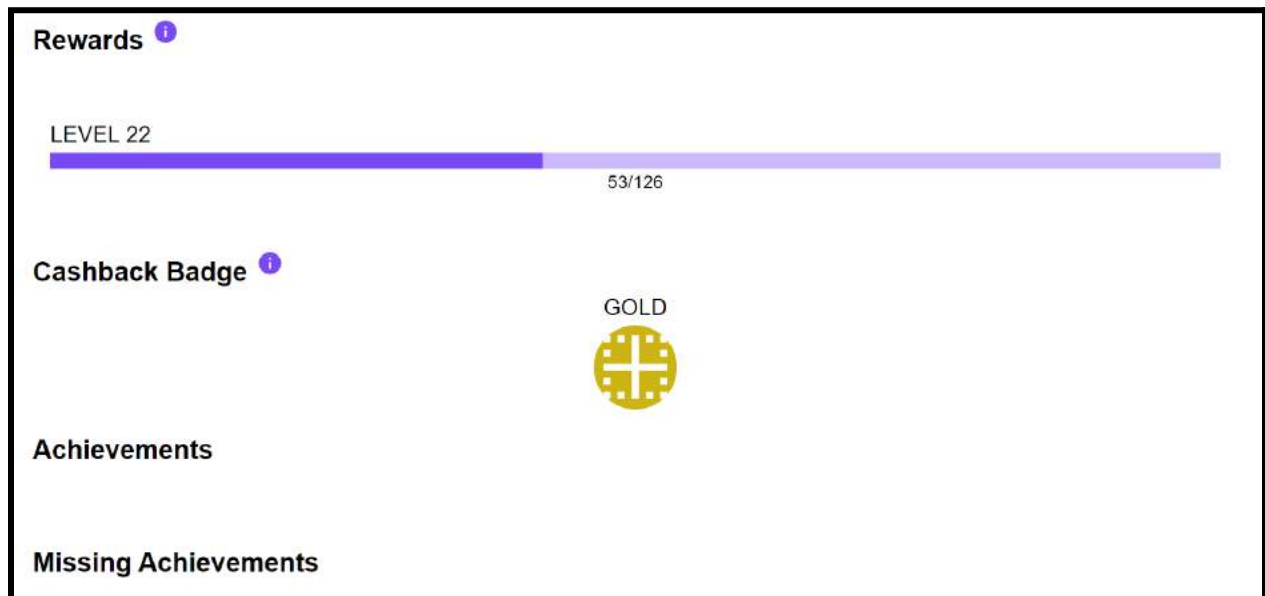
We level up and it is reflected on our reward page, thus achieving the level up objective.

Third-Party Functionalities

MongoDB Atlas

Our project uses MongoDB Atlas to store our data. We use the free "M0 Sandbox" cluster tier offered by MongoDB, which has a storage limit of 512MB. If our site were to include more product listings and users, and exceeded the limit, MongoDB provides the option to upgrade our cluster tier for a cost, to obtain more storage.

There is also a limit of 100 databases, 500 collections, and 100 active connections at a time, which is suitable for our project, but may affect it, if it were to be deployed to production for more users.

M0 free clusters can only be deployed in a subset of the regions available in higher tiered plans. This does not affect our project, as our cluster is currently stored in the Sydney region.

The "M0 Sandbox" cluster tier is a shared cluster and memory configuration, storage size configuration, backup and certain database operations are also not available in this tier. These could affect performance if our project was deployed to production and scaled up. However, these do not affect our project requirements currently.

The MongoDB Cloud Terms of Service states that MongoDB can suspend or terminate our access to usage of their service at any time, for reasons such as unlawful use. Our project does not engage in any illegal activity.

Gmail

Our project uses Nodemailer to send emails from an "admin" Gmail account. A free Gmail account is able to send a maximum of 500 emails per day, which is suitable for our project at its current scale. Gmail offers the option to convert to a paid account, which increases the maximum to 2000 emails per day.

IGDB API

The IGDB.com API is free for non-commercial usage under the terms of the [Twitch Developer Service Agreement](#).
Our project uses IGDB API endpoints such as "`https://api.igdb.com/v4/genres`", "`https://api.igdb.com/v4/companies/`", "`https://api.igdb.com/v4/games/`", and "`https://api.igdb.com/v4/covers/`" to read the data for available genres, companies, games, and cover images of the games, respectively. We used These endpoints to read data for 250 random games from the IGDB database and fill them in our game database. Using these endpoints, we managed to read full data for our games including name, price, companies, genres, summary, first release date, and cover image which is available on our website.

npm

Both the frontend and backend of our project uses the npm package manager. The npm registry is an online database of packages. Our project abides by the terms of acceptable use; it does not include content that is illegal, offensive or otherwise harmful. We do not redistribute any of the packages.

Frontend Packages

Package	Description	Use
typescript	Adds optional types to JavaScript.	Our frontend uses typescript. Static typing helps with code readability, and function reuse, especially when working in a team. It also helps find bugs during compilation, to help with debugging.
react	JavaScript library for building user interfaces.	Used for building our UI components. We also used Create React App to set up our frontend. Their licensing allows private use, commercial use, modification and distribution.
@types/react	Type definitions for react.	
react-scripts	Package includes scripts and configuration used by Create React App.	Used for scripts to start our app.
react-dom	React package for working with the DOM.	Used to render our app to the DOM.
@types/react-dom	Type definitions for react-dom.	
react-router-dom	React package which contains bindings for using React Router in web applications.	Used to handle routing in our app.
@types/react-router-dom	Type definitions for react-router-dom.	

@mui/material	Customisable component library of React components.	Used for our UI styling and to provide a consistent design throughout our app.
@mui/icons-material	Package with the Google Material icons packaged as a set of React components.	
react-icons	Includes popular icons for use in React projects.	
axios	Promise based HTTP client for the browser and node.js	Used to send HTTP requests to our backend.
moment	JavaScript date library for parsing, validating, manipulating, and formatting dates.	Used to track time before users can request to have another account verification or account recovery email sent to them.
jimp	Image processing library.	Used to scale uploaded profile images to set pixel dimensions, for improved performance.
react-easy-crop	React component to crop images.	Used for image cropping and resizing, when users upload a profile image.
react-google-captcha	React component for Google reCAPTCHA v2.	Used for CAPTCHA verification when redeeming and withdrawing store credits.
@types/react-google-captcha	Type definitions for react-google-capcha.	

Backend Packages

express	Framework for Node.js	Provides server side logic for our application.
express-jwt	Module which provides Express middleware for validating JWTs.	Used to check authorisation for requests sent to the backend.
cors	Package for providing	Used to allow our backend

	middleware that can be used to enable CORS with various options.	server to be accessible by our frontend.
dotenv	Module that loads environment variables from a .env file into process.env.	Used to access passwords, keys, secrets and other variables stored as environment variables.
mongodb	The official [MongoDB](#) driver for Node.js.	Used to create ObjectId values for searching the database by ObjectId.
mongoose	[MongoDB](#) object modelling tool designed to work in an asynchronous environment.	Used for connecting to our database and data modelling.
jsonwebtoken	An implementation of JSON Web Tokens.	Used to sign a JSON Web Token on user log in, for user authentication on future requests.
crypto-js	JavaScript library of crypto standards.	Used for AES encryption and decryption, when storing passwords and checking passwords.
nodemailer	Module for sending emails from Node.js.	Used to send account verification, account recovery links to the user. Also used to send confirmation when users redeem store credits and admins withdraw store credits.
python-shell	Module to run Python scripts from Node.js.	Used to run our search system and recommendation system algorithms.

Python Modules

All modules are under licenses which allow free commercial use, private use, modification and distribution.

Module	Description	Use
requests	Module for calling HTTP requests.	Used to call GET & POST APIs in python scripts.
bson	Module to encode images into binary format.	Used to encode images of the games into binary format.
re	Module to create regular expressions.	Used to run queries using regular expressions on MongoDB.
pymongo	Module to connect to MongoDB cloud.	Used to connect to MongoDB cloud and run queries on Database.
sklearn	Module containing Machine Learning tools and algorithms.	Used to run KNN algorithm in the recommendation system and search results.

Implementation challenges

Filling database with games from IGDB API

The first challenge for us was to collect data and add some games to our database. We decided to use IGDB API endpoints which is the Twitch database. The first difficulty was that, in order to access any API endpoint, a secret key needed to be provided which meant there was no publicly available API. Upon our connection to the API endpoints and reading the games, we realised that the default quality of the images are not sufficient and we need to improve their quality. Doing more research on IGDB's API documentation, we figured out that there is a url provided for each game containing its image where the last part of the url is a variable mentioning the size/quality of the requested image. This way, we could improve the images used in our website and enhance the user experience.

Finding the games similar to searched string

Another challenge was trying to find games related to the string searched by the user. The default autocomplete feature only finds the games containing the searched name as a substring and it might be very limited sometimes. For example, if the user searches for the word "Soccer" they probably also would like to see other sports related games as the result of their searched string. The way we managed to handle this part was that, upon user's search, the list of games containing the searched string is returned. Then each game is considered as a binary string where each bit shows for each genre if this game belongs to it or not. Finally, we apply the K-Nearest-Neighbors algorithm to find the most similar games to the autocomplete result and return them as the result of search history. As an example, searching the word "Soccer" results in not only soccer games, but also some other sports related games.

Defining similarity between games and recommendation system

Another challenge for us was coming up with a method that lets us recommend games to users. We thought that the recommendation system should consist of two main parts; one of them based on search history of the user, and another one based on the ratings given by the user and their similarity to other users' opinion. Then we needed to define the definition of "similarity" between two different games and come up with something that tells us how similar two games are. We decided that genres could be the best feature for defining similarity between games; so we considered each game as a binary string in which bit number i shows whether or not this game is of genre number i. Then, we apply the K-Nearest-Neighbors algorithm which is a popular Machine Learning algorithm using python modules and finally, we find the most similar games and recommend them to users.

User Authentication
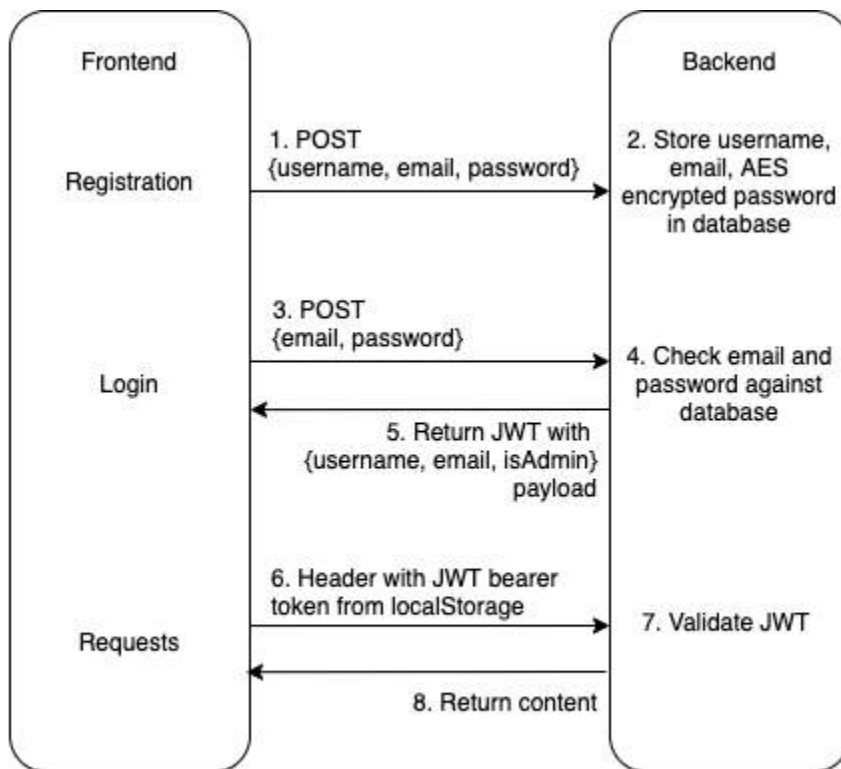
*Input validation is not included in this diagram

Our website includes a registration/login system and our database holds multiple users. We use 256-bit AES encryption to secure passwords. We return a signed JWT token on successful login. We validate the token with its signature for endpoints with restricted access, to prevent unauthorised users from accessing certain content.

Storing and Retrieving Images

Image Storage

↓

Upload Image
- Used FileReader object
- Read content as base64 string

↓

Crop / Resize Image
- Used Canvas element
- Return as base64 string

↓

Convert to Binary Data
- Remove MIME type
- Read base64 data into Buffer

↓

Rescale Image for Performance
- Used jimp to scale pixel dimensions to 300px by 300px
- Return data as Buffer

↓

Store Image
- Store image in MongoDB as Buffer type

Image Retrieval

↓

Retrieve Image
- Retrieve image from MongoDB as Buffer type

↓

Display Image
- Convert image back to bas64 string and add MIME type

Storing and retrieving images was a challenge, as we were initially unfamiliar with image manipulation. It involved a lot of research to understand the best image format for each step, and the different tools we could use.

Initially, profile images were stored in base64 format, which resulted in slow performance due to the large size of base64 data. This meant that POST and GET requests to the backend, involving the base64 image data, were slow. This was solved by converting the image to binary data, when it was stored.

Another issue was that uploaded images with high resolution took a longer time when it was retrieved and displayed. This was solved by using jimp, an image processing library, to rescale the image pixel dimensions, before it was stored.

Timer
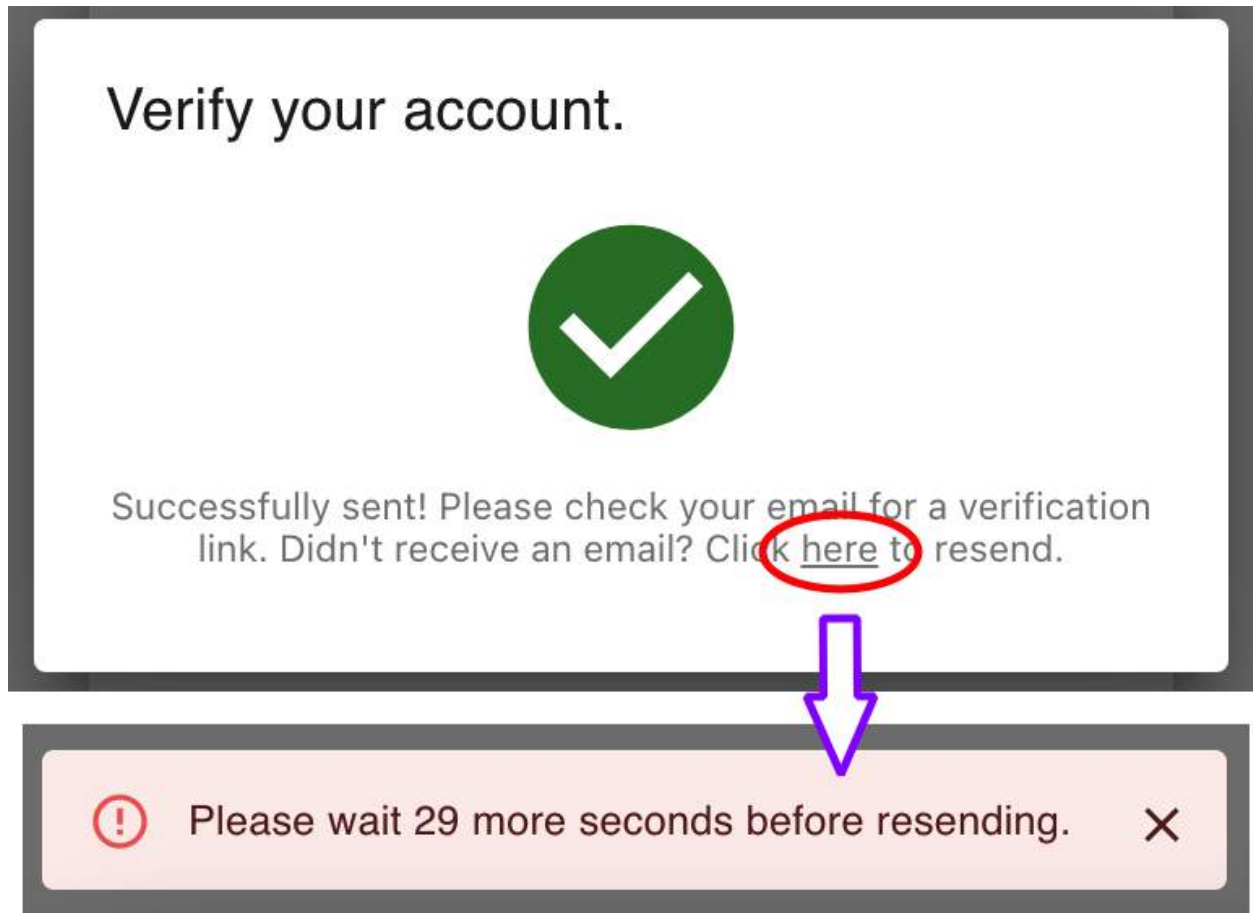
Some of our features required a "timeout" feature for both a better user experience, and to prevent abuse from users possibly spamming a request. For example, when an account verification email or an account recovery email is sent, there is a timeout of 30 seconds after the email is successfully sent, before the user can request to have the email reset to them. This allows time for the user to check their email.

In order to implement this, we wanted to have a timer in the background which did not interrupt the functionality of the rest of our website, so that users could still interact with the website as usual. This involved research to figure out the best way to do this, and was a challenge as we were unfamiliar with date/time manipulation. We decided to use moment to calculate and track the time when the next request could be sent.

```
setNextUpdate(moment().add(30, "s"));
setStatus("success");
```

On the next request, we would check the time, to see if the user can make a request.

```javascript
var timeNow = moment();
if (timeNow.isBefore(nextUpdate)) {
  var difference = nextUpdate.diff(timeNow, "s");
  alertError(
    `Please wait ${difference} more seconds before resending.`
  );
```

Reward Calculations

Another non trivial function was the calculation of rewards. The gist of the rewards is that for every $100 (cashBackThreshold) spent, depending on their badge they earn x back. The function itself relies on another function, namely a level information function. That function relies on a modified pokemon level experience formula, where the exp to next level is:

```javascript
Math.round((20 * ((level + 1) ** 1.1)) / 5)
```

As a result we technically only needed to retain total experience points and avoided needing to keep records of both levels and experience which would be cumbersome. The level information is mainly a for loop, which calculates the total level of the user, experience left to the next level and the badge of the user.

Using this function, it would provide an easy way to compare the current and previous state of the user and thus display the proper level up/reward notification.

So in the reward summary we call the level information function as the previous state. A cashback multiplier is then calculated using the function

```javascript
Math.floor(moneySpent / cashBackThreshold)
```

With the money left as the modulus of the cashBackThreshold. We then check if the money left qualifies us for another cashbackMultiple depending on how much left over that has been spent last time.

Using this cashMultiple, we then calculated their reward depending on their badge we calculated prior.

Once we add experience to their account, we call level information again. Now having both the before and after state of experience/level/badge information we can craft our reward message including the cashback amount, experience earnt and if they levelled up or increased their badge.

User documentation/manual
How to build, setup, configure, and use your system and functionalities on VLab.

This guide assumes you've downloaded the project and navigated to that directory.

Installations

1. You must install the Node.js (currently 16.13.0). Follow https://nodejs.org/en/ for instructions.
   a. You may need to update Node.js if on Vlab, do so by running

```
nvm install 16.13.0
```

2. You must install the Yarn package manager (currently 1.22.15, further information at https://classic.yarnpkg.com/lang/en/docs/install/#windows-stable). To install run the following in the terminal:

```
npm install --global yarn@1.22.15
```

3. Run bash script in main folder

```
chmod +x install.sh
sh install.sh
```

Backend

1. Navigate to backend folder

```
cd backend
```

2. Install dependencies

```
yarn
```

3. Start backend

```
yarn start
```

Frontend

1. Open a new terminal and navigate to frontend folder

```
cd frontend
```

2. Install dependencies

```
yarn
```

3. Start frontend

```
yarn start
```

4. Open a browser and navigate to http://localhost:3000/ if it has not opened already

Troubleshooting

If app does not start attempt following attempts:
1. Logout and login and restart frontend/backend apps
2. Check env files have correct ports
3. Update Node as seen in step 1

References

- *MongoDB 2021*, accessed 1 October 2021,
  https://www.mongodb.com/cloud-terms-and-conditions

- *MongoDB 2021*, accessed 1 October 2021,
  https://docs.atlas.mongodb.com/getting-started/
- *Npm 2021*, accessed 15 November 2021,
  https://docs.npmjs.com/policies/npm-license
- *Google 2021,* accessed 15 November 2021,
  https://support.google.com/a/answer/166852
- *Valve Corporation 2021*, accessed 1 October 2021, https://steamcommunity.com
- *Epic Games 2021*, accessed 1 October 2021,
  https://www.epicgames.com/store/en-US
- *Ubisoft 2021*, accessed 1 October 2021, https://ubisoftconnect.com/en-US
- *Internet Game Database 2021,* accessed 3 October 2021 , https://www.igdb.com/